

Document made available under the Patent Cooperation Treaty (PCT)

International application number: PCT/JP05/004452

International filing date: 14 March 2005 (14.03.2005)

Document type: Certified copy of priority document

Document details: Country/Office: JP
Number: 2004-132206
Filing date: 30 March 2004 (30.03.2004)

Date of receipt at the International Bureau: 28 April 2005 (28.04.2005)

Remark: Priority document submitted or transmitted to the International Bureau in compliance with Rule 17.1(a) or (b)



World Intellectual Property Organization (WIPO) - Geneva, Switzerland
Organisation Mondiale de la Propriété Intellectuelle (OMPI) - Genève, Suisse

日 本 国 特 許 庁
JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出 願 年 月 日
Date of Application: 2 0 0 4 年 3 月 3 0 日

出 願 番 号
Application Number: 特 願 2 0 0 4 - 1 3 2 2 0 6

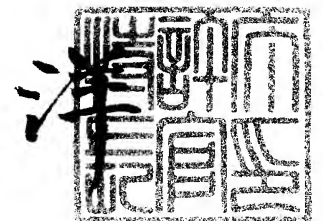
パリ条約による外国への出願
に用いる優先権の主張の基礎
となる出願の国コードと出願
番号
J P 2 0 0 4 - 1 3 2 2 0 6
The country code and number
of your priority application,
to be used for filing abroad
under the Paris Convention, is

出 願 人
Applicant(s): 藤 田 修

2 0 0 5 年 4 月 1 3 日

特許庁長官
Commissioner,
Japan Patent Office

小 川



【書類名】	特許願
【整理番号】	PAT04A0001
【提出日】	平成16年 3月30日
【あて先】	特許庁長官 今井 康夫 殿
【発明者】	
【住所又は居所】	奈良県奈良市朱雀3丁目15番地1号5－407号室
【氏名】	藤田 修
【特許出願人】	
【住所又は居所】	奈良県奈良市朱雀3丁目15番地1号5－407号室
【氏名又は名称】	藤田 修
【電話番号】	0742-72-3557
【提出物件の目録】	
【物件名】	特許請求の範囲 1
【物件名】	明細書 1
【物件名】	図面 1
【物件名】	要約書 1

【書類名】 特許請求の範囲

【請求項 1】

複数の成分からなるデータを記憶するデータ記憶装置と、重要度の高い成分を選別する重要成分選択装置と、各成分に関してプログラムされた演算処理を行う演算処理装置とで構成されるものであって、重要成分選択装置はある成分の重要度とそれがどの成分であるかを表す成分識別子を一組として、その複数組を重要度が高いものから順位付けして記憶する機能と重要度が最も高い成分の成分識別子を優先して出力する機能とを備え、演算処理装置は重要成分選択装置が指し示す重要度の高い成分に係る成分について演算処理し、その結果に従ってデータ記憶装置の成分のデータを更新するとともに、変更された成分に関する重要度を再評価して、その更新値を重要成分選択装置に送出する機能を備え、各装置はこれらの機能をそれぞれ同時並列に実行可能であることを特徴とする重要成分優先計算方式ならびに装置。

【請求項 2】

上記第 1 項の重要成分選択装置において、複数のレジスタと各レジスタに付属する比較器と、レジスタへのデータ入出力と各レジスタ間のデータのシフト動作を制御する制御回路を備え、各レジスタのデータは重要度データと成分識別子を含み、重要度の大きさの順にデータを整列して保持するとともに、最重要データを出力し、その最重要データが読み出されると、残りの各データの重要度の順位を 1 つ上げる機能を備え、各比較器は対応する各レジスタの重要度データ部と入力データの重要度データ部とを大小比較するとともに、対応する各レジスタのデータの成分識別子と入力データの成分識別子とを一致比較し、入力データ制御回路はその比較結果を基に、成分識別子について入力データと同じものがどのレジスタにも存在しない場合は、入力データよりも重要度が高いデータと低いデータとの間の順位に入力データを追加挿入し、入力データの成分識別子と同じ成分識別子を保持するレジスタが存在し、かつ、そのレジスタの重要度データが入力データの重要度よりも低い場合は、そのレジスタのデータを削除するとともに、入力データよりも重要度が高いデータと低いデータとの間の順位に入力データを挿入することを特徴とする重要成分優先計算方式ならびに装置。

【請求項 3】

上記第 2 項の重要成分選択装置において、各レジスタの重要度データ部のデータに対して演算処理を実行する演算回路を付加したことを特徴とする重要成分優先計算方式ならびに装置。

【請求項 4】

上記第 1 項のデータ記憶装置において、複数の成分データのの一つ一つに対して複数の候補値データを保持し、演算処理装置ではその複数の候補値の統計量をその成分の代表値とみなして各成分データについてプログラムで指定された演算処理を行い、各成分データの新しい更新値を算出し、その更新値と最も大きく異なる候補値をその更新値で置き換えるとともに、成分データの複数の候補値のばらつきの大きさをその成分データの重要度とみなして、重要成分選択装置において候補値のばらつきの大きさの順位に基づいて重要成分を選択することを特徴とする重要成分優先計算方式ならびに装置。

【書類名】 明細書

【発明の名称】 重要成分優先計算方式ならびに装置

【技術分野】

【0001】

高速なデジタル計算によるシミュレーション、特に、主として局所的相互作用が重要となる物理モデル、ネットワーク上の信号伝搬、高分子の立体構造予測などを対象とした高速シミュレーションを低コストで効率よく実行するための装置に関するものである。

【背景技術】

【0002】

従来、物理現象や社会現象など多数の成分からなるシステムのシミュレーションにおいては、システムの時間変化やエネルギー最小化、自己矛盾性の緩和などを計算するために、すべての成分の状態値とそれに関する条件式や状態遷移式を順番に逐一計算し、それを何度も繰り返して、状態値を少しずつ更新することにより、正しい数値を求めることを行なっていた。そのため、全成分数のべき乗に比例する程の計算時間を要し、成分数が膨大になると計算時間がかかり過ぎて結果を得る事が困難になるという欠点があった。

【0003】

特に、物理系の時間変化のシミュレーションの場合には、一つの時刻における全ての状態値を決めるためだけでも、エネルギー最小化や自己矛盾性の緩和計算が要求され、各状態値の修正計算を何度も繰り返して行なう必要があった。そのような計算を各時刻毎に繰り返して計算するため、膨大な計算時間を要した。全ての成分が大きく変動するようなシステムであったり、非常に精密な計算結果を得るためのものである場合など、どうしてもそれだけの計算時間が必要となることもあるが、そうではない場合においてさえも、従来の計算方式ではいつも同じ程度の計算時間を要するという欠点があった。

【0004】

探索や最適化問題の計算機による解法においては、分枝限定法のように、成分の全てを順番に計算せずに、計算の優先順位を表す成分リストを設けて、このリストに従って計算を進める手法がある。しかし、この手法は正確な演算を目指したもので、成分データの計算過程と優先順位リストの更新はソフトウェアのアルゴリズムとして逐次的に処理されるため、リストの更新に時間がかかり、その分、全体の処理時間も長くなるという欠点があった。また、このようなソフト的な工夫を前述の物理シミュレーションのように各成分の演算処理が比較的単純な場合に適用しようとする、各成分の演算処理コストに比べて優先順位の計算にかかるコストの比率が大きく、計算効率があまり改善されないという欠点があった。

【発明の開示】

【発明が解決しようとする課題】

【0005】

例えば、実際の物理現象や社会現象などでは、ある時刻においてほんの一部の成分のみが変化し、それ以外の大多数の成分はほとんど変化しない場合がある。また、もともとシステム内外のノイズが大きい場合や、外部条件が種々雑多で統計的、確率的に取り扱うべき場合など、もともとの数値データの精度が低いものに対して必要以上に精密な計算をすることが無意味な場合もある。このような場合、従来技術の計算処理においては最終結果にあまり影響を与えないような無駄な計算の繰り返しが多く含まれていた。本発明はこのような無駄な計算を省き、シミュレーション計算を低コストで効率よく行なうためのものである。

【課題を解決するための手段】

【0006】

本発明では計算の重要度の高い成分を選択し、重要度の高い成分から優先的に計算を行い、計算対象となった成分の重要度を再評価して、その結果をその後の重要成分の選択に反映させる。重要度の低い成分に対する計算は後回しにするか、または一定の基準で無視することにより、できるだけ早く効率よく概算結果を獲得し、その後は必要に応じて時間

をかけて計算精度を向上させる。

【0007】

そのため、各成分の値の計算以外に各成分の重要度を計算する過程を加え、それを処理するための専用装置として重要成分選択装置を設け、その装置によって各成分をその重要度に応じて順位付ける。また、各成分の状態値の計算においては重要成分選択装置の指示する最重要成分を優先して計算処理を行なう。各成分の状態値の計算を独立に行なってもよいとする計算モデルに対しては、一つの共有メモリにアクセスできる複数の演算処理装置を設け、演算処理を同時並列化し、さらなる高速化を実現する。なお、演算処理装置と重要成分選択装置とはそれぞれ同時並列に動けるようにしておく。演算処理装置は一つの成分に関する処理が終わるとすぐに次の重要成分を読み出してその処理を実行するようにしておく。重要成分選択装置もその内部処理は独立に動作できるようにしておく。応用事例によっては、一つの成分のみが連続して最重要成分となってその計算処理が繰り返されることを防ぐために、重要度の更新において一定の時間遅れを設定したり、可変に調整できるようにしておく。

【0008】

重要成分選択装置において最高速の処理を要求される計算モデルに対しては、重要成分の順位付け計算をハードウェアで実現する。具体的には、データの重要度を表す重要度データ部とそれがどのデータのものであるかを指し示す成分識別子を保持するレジスタを設け、それを複数個、上位から下位へ順番に並べる。各レジスタは隣接レジスタ同士でデータをシフトできるようにしておく。また、各レジスタに付属する比較器を置き、各レジスタのデータと入力データを同時に比較して、その結果に従って入力データを重要度の順位を満たす位置に挿入する。ただし、入力データの成分識別子と同じものがすでに記憶されている場合には、入力データの重要度の方が記憶されているものより大きい場合のみ挿入され、重要度の値が低いデータは削除する。初期値として整列した状態に設定しておけば、その後に順次入力されるデータは常に重要度の順位に整列して保持される。また、同じ成分識別子のデータが2個以上記憶される事はない。このハードウェアを使えば、入力データの挿入動作と出力動作を1マシンサイクルで同時に処理できるので、重要度によるデータの整列を高速に実行することができる。

【0009】

重要成分選択装置において、重要度の高いデータは次々に出力されて処理され、重要度の低いデータは最下位レベルの方向へ押し出されていくが、重要度が中程度のデータが長い期間残留し、なかなか演算対象にならない場合が出てくる。そのような古いデータの残留を防ぐ必要がある場合には、各レジスタの重要度データに対する演算処理を施すことができる演算器を設ける。この演算器を利用して各データに対して定期的に一定値を加算またはビットシフトや乗算することにより、各データの順序関係を壊さないように重要度データを大きな値に書き換えると、古いデータの重要度が増して演算処理がなされるように調整できる。逆に重要度データを定期的に小さな値に書き換えることにより、古いデータをできるだけ早く消去するように調整することもできる。

【0010】

変化の少ない成分に関する無駄な演算処理を省略するため、記憶装置において複数の成分データのの一つ一つに対して複数の候補値データを保持するものとする。演算処理装置ではその複数の候補値の統計量を代表値として成分データの値とみなし、各成分データについて計算処理を行ってその更新値を算出し、その更新値と最も大きく異なる候補値をその更新値で置き換える。この場合の統計量としては平均値や中央値などを用いる。そして、重要成分選択装置では候補値のばらつきの大きさをその成分データの重要度とみなして、候補値のばらつきの大きさの順位で重要成分を選択する。候補値のばらつきを重要度とみなす理由は、そのことによりシミュレーションにおいて成分データの値が十分に収束していない状況や時間変化が大きい状況を表すことができ、変動の少ない成分データよりも変動の大きい成分データの計算処理を優先させることができるためである。これにより、計算の最終結果に対して影響の少ない計算過程を省略することができ、より高速に概

算結果を得ることが可能になる。

【0011】

なお、一部の成分のみが重要度が高くなり、それらの計算処理に偏る可能性がある場合には、他の成分が大きな重要度をもつように調節する機能を入れておく。例えば、演算処理のプログラムの一部に、乱数によって選ばれた成分に大きな重要度を与えて重要成分選択装置に送る処理を加える方法、一定時間ごとに全成分を計算する処理を割り込ませる方法などを加えてソフトウェアで調整する。あるいは、重要成分選択装置において重要度をすべて一斉に最大値にセットし、そこに記憶されている成分の全てを順番に計算処理する方法などを用いる。

【発明の効果】

【0012】

シミュレーションの対象となる物理現象や社会現象において、ノイズが大きいとか初期値依存性が小さいなど、高精度の演算を必要としない場合に、重要度の高い計算を優先して、無駄な計算を無視または後回しにすることにより、少ない計算資源で効率良く概算結果を得ることができ、計算時間が短縮され、計算コストが削減されるという効果がある。

【発明を実施するための最良の形態】

【0013】

以下、本発明の実施の形態を図1～図20に基づいて説明する。

【0014】

本発明の装置の第一の実施形態の基本ブロック構成図を図1に示す。また、各ブロックの動作の概要を図2に示す。本装置は主としてデータ記憶装置11と演算処理装置12と重要成分選択装置13とで構成される。データ記憶装置11は成分1から成分NまでのN個のデータを記憶する。演算処理装置12は各成分に対応した処理内容を実行する。処理内容はプログラムで設定する。重要成分選択装置13は、各成分の重要度とそれが1からNまでのどの成分のものを表す成分識別子を記憶し、その中で最も重要度の高い成分を選択する機能を有する。重要成分選択装置13に一つの成分の重要度と成分識別子が入力されると、その度に、重要度の大きさの順序に重要度と成分識別子を整列させる。この場合の処理は、一つの成分が入力される時にはその前にすでに入力された成分は整列している状態なので、重要度の大小関係を満たす位置を探してそこに挿入する処理となる。なお、この実施形態においては、成分識別子として成分番号を用いている。

【0015】

本装置を動作させるには、まず各成分データに初期値を設定し、演算処理装置12は初期化ルーチンを実行する。初期化ルーチンとしては全成分に関する演算を順番に逐次実行する方法、一部成分についてランダムに演算を実行する方法などにより、演算に必要な成分の値をデータ記憶装置11から読み出し、各成分に関する演算処理を実行して、成分の更新値を求める。また、各成分の重要度を計算し、重要成分選択装置13に各成分の重要度と成分識別子（成分番号）を入力する。そして、各成分の更新値をデータ記憶装置11に記憶させる。

【0016】

初期化ルーチンの代わりに、重要成分選択装置13のM個のレジスタに各成分に対応する成分識別子と重要度データを初期値として代入しておいてもよい。それらの重要度全てに最大値を設定しておけば、初期値として設定した成分すべてが優先され、それらが成分識別子（成分番号）の並び順に従って順番に演算処理される。

【0017】

初期化処理後の主要な実行処理内容の概要を図2に示す。演算処理装置12はまず重要成分選択装置13から重要度が最大の成分識別子（成分番号）を読み出す。その成分番号をiとすると、次に演算処理装置12は成分iをデータ記憶装置11から読み出し、成分iに関する処理を実行する。具体的処理内容は応用目的に応じて演算処理装置12のソフトウェアとして最適なプログラムを組むことを可能としておく。この処理において、成分iのみならず、その他の一つ以上の成分、例えば成分jについても更新することができるも

のとし、更新する成分についてはその成分の重要度を計算する。この後、更新する成分に関して、その重要度と成分番号を重要成分選択装置 1 3 に送出し、成分の更新値をデータ記憶装置 1 1 に送出する。この一連の処理が終わると、直ちに重要成分選択装置 1 3 から最重要成分を読み出して同様の処理を繰り返す。演算処理装置 1 2 はこの一連の処理をプログラムで設定された終了条件に達するまで続ける。

【0018】

重要成分選択装置 1 3 は演算処理装置 1 2 によって最重要度の成分識別子（成分番号）が読み出されると、そのデータを削除し、その他の成分識別子の順位をそれぞれ一つ上げる。演算処理装置 1 2 から重要度と成分番号の一組の入力データが入力されると、その度毎に重要度の順序に従うように整列させる。整列の順位が k 番目のデータの重要度の値を $P(k)$ で表すと、整列により $P(k) \geq P(k+1)$ となっているので、入力データの重要度を P_{in} とすると、 $P(k) \geq P_{in} > P(k+1)$ となる k 番目の位置を探し、 $k+1$ 番目以降のデータのすべてについて順位を一つ下げて、入力データを $k+1$ 番目におく。最下位の M 番目にあったデータは捨てられる。ただし、もし記憶データの中に入力データの成分識別子と同じ成分識別子をもつデータがある場合は、両者の重要度を比較して、入力データの重要度の方が小さいときには挿入はしない。逆に入力データの重要度の方が大きいときには上述のように挿入し、記憶されていた方のデータは削除する。この入れ替えのため、記憶されていたデータよりも重要度の低いデータの順位は変わらない。

【0019】

第 2 の実施形態を図 3 に、その主要な動作概要を図 4 に示す。第 1 の実施形態と異なる点は、データ記憶装置 2 1 に成分データとともに各成分データが選択されたときに実行すべき処理事項を記憶させ、演算装置 2 2 においてその情報を利用して実行処理内容を制御するようにしておく点である。こうすることにより、成分データの内容に応じて処理事項を適宜変更する必要がある場合に、効率良く演算処理が出来る。処理事項として記憶すべきデータは、例えば、同時に計算処理すべき他の成分データの識別子のリストであるとか、その成分が属するクラスの名称または識別子などである。

【0020】

第 3 の実施形態を図 5 に示す。重要成分選択装置 3 3、演算処理装置 3 2、データ記憶装置 3 1 についてそれぞれ複数個を並列化して、各装置が同時並行してそれぞれの処理を実行することで、全体の演算処理を高速化する。複数個の演算処理装置 3 2 と複数個の重要成分選択装置 3 3 との間のデータ転送はデータ転送制御回路 3 5 によって制御する。また、複数個の演算処理装置 3 2 と複数個のデータ記憶装置 3 1 との間のデータ転送もメモリ入出力制御回路 3 4 によって制御する。なお、演算処理装置 3 2 を複数個にし、それぞれが、一つの共通の重要成分選択装置に時分割でアクセスするようにしてもよい。

【0021】

データ転送制御回路 3 5 の制御の方法としては様々な方法が適用可能である。例えば、データ転送制御回路 3 5 にスイッチングマトリクスを利用し、任意の演算処理装置 3 2 から任意の重要成分選択装置 3 3 へデータ転送できるようにしておき、その接続の対応関係を適宜変更可能なようにしておく。ただし、演算処理装置 3 2 から重要成分選択装置 3 3 へデータ転送する場合のみ接続の組み合わせを替え、重要成分装置 3 3 から演算処理装置 3 2 へのデータ転送では常に同じ組み合わせの接続に固定しておいてもよい。逆に、演算処理装置 3 2 から重要成分選択装置 3 3 へのデータ転送の接続の組み合わせを固定し、重要成分装置 3 3 から演算処理装置 3 2 へのデータ転送の接続の組み合わせを変化させてもよい。また、一つの重要成分装置 3 3 への入力データが同時に多数集中する可能性がある場合には、各演算処理装置側に F I F O バッファレジスタを設置して、演算処理装置 3 2 に無駄な待ち時間が生じないようにする。

【0022】

メモリ入出力制御回路 3 4 の制御は分散共有メモリのための入出力に関する従来技術を使用する。さらに、データの読出し効率を上げる場合には記憶装置のデータ出力をマルチポートにして異なる複数のデータを同時に出力できるように並列化する。このためのデー

タ記憶装置34の実施形態を図6に示す。複数の記憶素子38のデータを複数のデータセクタ39で同時並列に読み出せるようになっている。この場合、書込みデータは一時的にFIFOバッファレジスタ36に保存し、その後書き込み制御回路37によって順番に記憶素子38に書込みを行なう。複数の演算処理装置から書込みデータが同時に多数集中する可能性がある場合には、さらに各演算処理装置側にFIFOバッファレジスタを設置して、演算処理装置に無駄な待ち時間が生じないようにする。

【0023】

重要成分選択装置をハードウェアで実現する場合の第4の実施形態における重要成分選択装置の基本ブロック構成図を図7に示す。この装置はデータ整列制御回路420とn個のデータを保持するn個のレジスタ410とからなる。各レジスタ410のデータを $R(1)$ 、 $R(2)$ 、 \dots 、 $R(n)$ と表すと、データ整列制御回路420はデータに含まれる重要度が最も高いデータを $R(1)$ に、次に高いデータを $R(2)$ に、というように大小の順番に従って、データを整列してレジスタに保持する。入力データがある場合はデータ整列制御回路420に制御信号としてPUSHを入力する。入力データを Din と表すと、入力データ Din の重要度 P_{in} が $R(n)$ の重要度 $P(n)$ より高ければ、適切なレジスタに入力データを挿入して、全体の大小の順序関係を保つようにレジスタのデータを並べ替える。また、出力からは $R(1)$ のデータを読み出すことができる。読出しの場合にはデータ整列制御回路420に制御信号としてPOPを入力する。すると、各データは順位を一つ上げ、 $R(k+1)$ は $R(k)$ へとシフトする。ただし、 $R(n)$ に対しては重要度が最低の値、たとえば、全ビットに0を入力する。なお、番号が小さい程、重要度が高いので、番号が小さい方のレジスタを上位のレジスタ、番号が大きい方のレジスタを下位のレジスタとみなす。

【0024】

データ整列制御回路420の詳細な構造を説明するため、 $n=4$ の場合の例を図8に示す。各レジスタ410に対応して比較器421とセクタ・演算処理器422を備えている。制御信号は各回路ブロックを制御するレジスタ制御信号と比較制御信号と選択・演算制御信号とからなる。レジスタ制御信号はレジスタのデータ取り込みを制御する。レジスタ k に対応する比較器 k は入力データ Din とレジスタ k のデータ $R(k)$ を比較して、その結果を出力する。セクタ・演算処理器 k はレジスタ k に保存すべきデータとして入力データ Din 、レジスタのデータ $R(k-1)$ 、 $R(k)$ 、 $R(k+1)$ の4つのいずれかを選択して出力することができる。どのデータを選択するかは比較器 k 、 $(k-1)$ 、 $(k+1)$ の比較出力によって制御される。また、セクタ・演算処理器 k は $R(k)$ のデータに対して加減乗除、ビットシフト等の演算処理を施した値を出力する。

【0025】

比較器421のさらに詳細な構成を図9に示す。入力データならびにレジスタに保持されるデータは重要度と成分識別子を含む。比較制御信号はデータに含まれる重要度と成分識別子の部分を抽出するための情報としてそれぞれのビット構成を設定するマスクデータを含み、大小比較および一致比較すべきデータを制御する。大小比較器423では入力データ Din の重要度 P_{in} とレジスタ k のデータの重要度 $P(k)$ を比較して、入力データの方が大きい場合、すなわち $P_{in} > P(k)$ のときに $G(k) = 1$ を出力し、それ以外の場合には $G(k) = 0$ を出力する。一致比較器424は入力データの成分識別子とレジスタ k のデータの成分識別子を比較して、両方が一致する場合に $S(k) = 1$ を出力し、それ以外の場合には $S(k) = 0$ を出力する。また、上位の比較器のうち一つでも $S(k) = 1$ となる場合は $LS(k) = 1$ を出力し、それ以外の場合には $LS(k) = 0$ を出力する。そのため、 $S(k) = 1$ の出力を下位の比較器すべてに伝達する回路が設けられている。すなわち、比較器 $(k-1)$ の出力 $SD(k-1)$ を比較器 k の入力 $SU(k)$ に入力し、その信号を $LS(k)$ として出力するとともに、 $SU(k)$ と $S(k)$ の論理和を $SD(k)$ に出力して、下位の比較器 $(k+1)$ の $SU(k+1)$ に接続する。

【0026】

レジスタ数が多い場合は、 $LS(k)$ 出力の下位比較器への伝播に時間がかかり、処理

速度を低下させる原因になることがあるので、伝播時間を短縮するため、信号の伝播回路にバイパスを設ける。バイパス回路を使用する場合の実施形態を図10に示す。3入力ORの入力の一つには入力端子B(k)を接続する。入力端子B(k)には(k-1)よりも上位の比較器hの出力SD(h)を接続する。ORの入力数を4以上にしてさらに上位の比較器と接続してもよい。

【0027】

図11はレジスタ数が19個の場合に3入力のORを利用したバイパス配線の実施形態を示す。LS(2)からLS(19)までの出力を生成するために論理素子ORがOR(1)からOR(18)まで18個使用されている。ただし、最上位の比較器には上位からの入力はなく、OR(1)の入力S(0)は常に0に固定する。従って、OR(1)を省略してS(1)を直接OR(2)とLS(2)に接続しても良い。バイパス配線については、全部のORをほぼ3等分して、上位から約2/3の位置にあるOR(13)を3入力とし、この一つの入力に上位から約1/3の位置にあるOR(6)の出力をバイパス配線として接続する。これにより、入力Sから出力LSまでの最長論理段数は18から12へと短縮される。更に最長論理段数を低減するためには、OR(1)からOR(6)までをブロックB(1,1)、OR(7)からOR(12)までをブロックB(1,2)、OR(14)からOR(18)までをブロックB(1,3)とし、各ブロック内でバイパス配線を行なう。各ブロックを上位と下位のほぼ同規模のサブブロックに分け、ブロックの最下位と中間のORを3入力としてそれぞれのすぐ上位のサブブロックをバイパスする配線を設ける。例えば、ブロックB(1,2)の場合、サブブロックB(2,3)とB(2,4)に分け、OR(9)とOR(12)を3入力としてそれぞれにOR(6)の出力とOR(9)の出力を接続する。ただし、最上位のサブブロックB(2,1)と最下位のサブブロックB(2,6)に対してはバイパス配線を設ける効果はほとんどないので省略している。サブブロックのバイパス配線を追加した結果、最長論理段数は8に改善される。

【0028】

レジスタ数が更に多い場合には図12に示すようにサブブロックB(i,j)内をさらにほぼ2等分してサブブロックB(i+1,2j-1)とB(i+1,2j)に分け、それぞれのバイパス線を設けることによりサブブロック内の最長論理段数は約半分に短縮される。この方法を一般的に適用して、最小のサブブロック内の論理段数をすべてNb、サブブロックの階層数(すなわちバイパス線の階層数)をMhとすると、バイパス線の総本数が $3(2^{Mh}-1)-2Mh$ で、最長論理段数が $2Nb+3Mh-2$ で、レジスタ数が $3Nb2^{Mh}+3(2^{Mh}-1)-Mh$ となる回路を構成する事が出来る。なお、図11の場合はNb=2、Mh=2であり、バイパス線の総数は5、最長論理段数は8、レジスタ数は19である。

【0029】

レジスタkに対応するセクタ・演算処理器kの詳細な構造を図13に示す。レジスタkに入力するデータを選択するデータセクタ425とレジスタkのデータR(k)に演算処理を施してその結果を出力する演算処理器426とデータセクタ425の出力と演算処理器426の出力のどちらかを選択する出力セクタ427から構成されている。データセクタ425は入力データDin、レジスタのデータR(k-1)、R(k)、R(k+1)の4つのいずれかを選択して出力する。選択・演算制御信号は出力セクタ427の選択制御を行うとともに、加数、ビットシフト数などの演算パラメータを含み、演算処理も制御する。制御信号はデータを入力処理するPUSH、データを出力処理するPOP、加算処理をするADD、MSB方向へのビットシフトSF TL、LSB方向へのビットシフトSF TRの5種類の動作を制御する。

【0030】

セクタ・演算処理器kの制御信号とセクタ・演算処理出力の関係を図14に表形式で示す。詳細条件として重要度と同一成分順位に関する条件により出力は異なる。各条件は比較器出力の組み合わせ論理で表される。ただし、記号*は無条件または論理値としてのドントケアを表す。Ksは入力データDinと同じ成分識別子のデータがレジスタに記

録されている場合のレジスタ番号を表す。すなわち $S(K_s) = 1$ である。入力データと同じ成分識別子のデータがレジスタに記録されていない場合はすべての k において $S(k) = 0$ で、 $LS(n) = 0$ となるので、便宜上、 $K_s = n + 1$ とみなしてよい。

【0031】

PUSHの場合、レジスタ k のデータ $R(k)$ の重要度の値 $P(k)$ よりも入力データ D_{in} の重要度の値 P_{in} が以上となる条件、すなわち、 $P(k) \geq P_{in}$ の場合は $R(k)$ を変更する必要がないので、セクタ・演算処理出力は $R(k)$ となる。この条件は比較器出力 $G(k) = 0$ のみで表され、他の比較器出力には依存せず、 $G(k-1) = *$ 、 $G(k+1) = *$ 、 $S(k) = *$ 、 $LS(k) = *$ である。条件が $P(K-1) \geq P_{in} > P(k)$ かつ $k \leq K_s$ というのは D_{in} の挿入位置の条件である。すなわち、 $G(k-1) = 0$ 、 $G(k) = 1$ 、 $G(k+1) = *$ 、 $S(k) = *$ 、 $LS(k) = 0$ で表され、この k に対してセクタ・演算処理出力は D_{in} となる。条件が $P_{in} > P(k-1) \geq P(k)$ かつ $k \leq K_s$ というのは D_{in} よりも重要度が低いデータで、かつ、 $P(k) \geq P(K_s)$ なので、 D_{in} の挿入により順位が一つ下がる $R(k)$ を意味しており、 $G(k-1) = 1$ 、 $G(k) = 1$ 、 $G(k+1) = *$ 、 $S(k) = *$ 、 $LS(k) = 0$ の条件で、一つ上の順位の $R(k-1)$ を $R(k)$ に代入するため、セクタ・演算処理出力は $R(k-1)$ となる。また、 $k > K_s$ の場合は D_{in} を挿入しても $R(K_s)$ が削除されるので順位が変わらない条件を表す。 D_{in} が挿入されない場合も順位は変わらないので、これは重要度の値によらず、 $G(k-1) = *$ 、 $G(k) = *$ 、 $G(k+1) = *$ 、 $S(k) = *$ 、 $LS(k) = 1$ の条件でセクタ・演算処理出力は $R(k)$ となる。

【0032】

POPの場合、 $P(k) \geq P(k+1) \geq P_{in}$ 、すなわち、 $G(k-1) = *$ 、 $G(k) = 0$ 、 $G(k+1) = 0$ 、 $S(k) = *$ 、 $LS(k) = *$ の条件において $R(k+1)$ は入力データの影響を受けないので、 $R(1)$ の読出し後に順位が一つ上がり、 $R(k+1)$ は $R(k)$ に移動する。すなわち、 $R(k)$ には一つ下位の $R(k+1)$ が代入されるので、セクタ・演算処理出力は $R(k+1)$ となる。 $P(k) \geq P_{in} > P(k+1)$ かつ $k < K_s$ 、すなわち、 $G(k-1) = *$ 、 $G(k) = 0$ 、 $G(k+1) = 1$ 、 $S(k) = 0$ 、 $LS(k) = 0$ の条件は D_{in} を挿入する条件でセクタ・演算処理出力は D_{in} となる。 $P_{in} > P(k)$ かつ $k < K_s$ 、すなわち、 $G(k-1) = *$ 、 $G(k) = 1$ 、 $G(k+1) = *$ 、 $S(k) = 0$ 、 $LS(k) = 0$ の条件は $R(1)$ を読出ししても D_{in} が挿入されるので順位が変わらないことを表す条件で、セクタ・演算処理出力は $R(k)$ となる。最後に、 $k = K_s$ すなわち $S(k) = 1$ または $k > K_s$ すなわち $LS(k) = 1$ の条件は他の条件によらず、この位置の k では上位において D_{in} の挿入があったとしても上位の $R(K_s)$ が削除されるので、 D_{in} の挿入の有無にかかわらず、 $R(1)$ の読出し後に下位の順位を一つ上げるため、セクタ・演算処理出力は $R(k+1)$ となる。

【0033】

演算処理は $R(k)$ に含まれる重要度 $P(k)$ に対して施される。ADDは選択・演算制御信号に含まれる演算パラメータ a を $P(k)$ に加算する。SFTLは a ビットだけMSB方向にビットシフトする。SFTLは a ビットだけLSB方向にビットシフトする。これらの処理により、重要度 $P(k)$ の値を変化させ、記憶されたデータが出力される可能性を高めたり、あるいは逆に削除される可能性を高めることにより、古いデータが長時間にわたってレジスタに留まることを防止することができる。

【0034】

次に、図7の重要成分選択装置の動作について以下に説明する。最初に全レジスタに一定値を記憶させ、新しいデータが入力されると、その入力データの重要度データと各レジスタの記憶データの重要度データとを即時に比較し、重要度データ部の値が入力データより大きいかまたは等しいレジスタの記憶データはそのままの状態とし、重要度データ部の値が入力データより小さいレジスタの記憶データはすべて下位方向へシフトさせ、記憶データが入力データより大きいレジスタ群と小さいレジスタ群との間に一つ空いたレジスタ

に入力データを挿入する。ただし、入力データの成分識別子と同じデータがレジスタに記憶されている場合にはこのレジスタに保持されている重要度が入力データの重要度より大きい場合には挿入されない。逆に入力データの重要度より小さい場合にはこのレジスタのデータを削除し、入力データを適切な位置に挿入する。同じ成分識別子のデータが重複して記憶されることはない。このようにして順次入力データを追加していくと、上位から下位へ重要度の高いものから低いデータへと整列して保持される。重要度が最高のデータは上位のレジスタから読み出され、その際、全レジスタを上位方向に一つシフトして、次に重要なデータを出力できるように準備する。入力データの挿入動作と出力動作を1マシンサイクルで同時に処理できるので、重要度によるデータの整列を高速に実行することができる。

【0035】

このような場合の動作の具体例を図15、16、17に示す。いずれも図14の規則に従う。図15はPUSHの場合で、入力データDin(Pin=60)をP(3)=60とP(4)=33の間に挿入するため、DinはR(4)に代入され、R(4)、R(5)はそれぞれR(5)、R(6)に移動する。R(6)の成分識別子555は入力データの成分識別子と同じで、重要度がPin=15と低いので、R(6)は削除される。R(7)、R(8)は変化しない。図16は同じ入力データがあるときのPOPの場合で、R(1)が読出された後に重要度が60以上のR(2)、R(3)はそれぞれ順位を一つ上げる。R(4)の重要度Pin=30は入力データより低いので、DinがR(3)に挿入され、R(4)の順位は変化しない。R(5)も変化しない。R(6)は入力データの成分識別子と同じで重要度が低いので削除されるため、下位のR(7)、R(8)がそれぞれR(6)、R(7)に移動する。最下位のR(8)はデータがクリアされる。これは重要度が最小値0で、成分識別子がNULLであることに相当する。図17はPOPの場合であるが、入力データの重要度がPin=7で、同じ成分識別子555を記憶するR(6)の重要度P(6)=15より低いので入力データが挿入される事はなく、R(1)の読み出し直後に全てのレジスタのデータが順位を一つ上げる。すなわち、 $k > 1$ についてR(k)はR(k-1)に移動し、R(8)はクリアされる。

【0036】

図18は変化の少ない成分に関する無駄な演算処理を省略するための成分データの表現方式に関する説明図で、本発明の第5の実施形態を示す。データ記憶装置51において、1つの成分に対して複数個の候補値データを保持しておく。例えば、i番目の成分y(i)の候補値として $x(i, 1), x(i, 2), \dots, x(i, m)$ のm個を保持する。演算処理装置52では、この複数の候補値の代表値をy(i)とみなして演算処理を行なう。代表値としては $x(i, 1), x(i, 2), \dots, x(i, m)$ の統計量として、例えば、平均値、中央値などを用いる。演算処理はプログラムによって制御される。この演算処理の結果、y(i)の更新値としてz(i)が得られると、候補値の中で更新値z(i)からもっとも距離の遠い候補値を更新値z(i)で置き換える。なお、更新された値を使って、再度代表値y(i)を計算しなおし、これをデータ記憶装置51に記憶しておけば、次の計算ですぐに利用できるもので、計算効率が良くなる場合がある。

【0037】

各成分の重要度としては、 $x(i, 1), x(i, 2), \dots, x(i, m)$ のばらつきの大きさを $\Delta x(i)$ と表し、これを成分iの重要度とみなす。ばらつきの大きさとしては、例えば、候補値 $x(i, 1), x(i, 2), \dots, x(i, m)$ の最大値と最小値の差の絶対値 $\max\{|x(i, u) - x(i, v)|\}$ や標準偏差などを用いる。成分データの候補値のばらつきが大きい成分は、演算処理による更新値の変動が大きい事を意味しており、シミュレーションなどにおいては成分データの値が十分に収束していない状況や時間変化が大きい状況を表している。ばらつきの大きさを重要度とみなす事により、変動の少ない成分データよりも変動の大きい成分データの計算処理を優先させて演算処理を進める事になる。これにより、変化の少ない計算過程を省略することができ、より高速に概算結果を得ることが可能になる。

【0038】

重要成分選択装置53で $\Delta x(i)$ が最も大きいとして成分 i が出力されると、演算処理装置52はプログラムに従って i 成分に関する演算処理を実行するが、必要に応じてプログラムにより i 成分と関連性の深い他の成分についても演算処理を行い、これを更新する。他の成分として例えば h 成分も演算処理すると、更新の結果として i 成分と h 成分の候補値を更新するとともに、演算処理装置52は i 成分の重要度と h 成分の重要度を重要成分選択装置53に送る。演算処理装置は、一つの演算処理ループを終了すると、直ちに重要成分選択装置53からその時点の最重要成分の出力を読み出して、次のループ処理に入る。もし重要成分選択装置53の処理時間に遅れがあったとしても、それを待たずに実行できるようにしておく。

【0039】

図19は成分データの表現方式に関する説明図で、本発明の第6の実施形態を示す。成分 i の値の変化に伴って、成分 i に関して演算処理すべき内容が変化する場合に、各成分に付属する情報として処理事項の情報を各成分毎にデータ記憶装置61に保持しておく。図19においては、処理事項を a 、 b 、 c などで表している。具体的には、例えば、成分 i の処理事項 h は成分識別子 h を表しており、成分 i に関する演算処理において参照する必要のある他の成分識別子である。この処理で成分 h の値を更新してもよい。あるいは、成分2の処理事項 d を成分2が属するクラスの名称または識別子名としてもよい。クラスに属している成分全てを同時に演算処理する必要があるときなどに用いる事が出来る。

【0040】

応用例として、多粒子系の物理学的シミュレーションの場合を以下に説明する。ある一つの粒子の位置座標を一つの成分として表し、各粒子に働く相互作用がある一定距離以内の粒子のみに働く物理系とする。粒子が運動する空間を複数の領域に分割し、各領域をそれぞれ一つのクラスとする。例えば、空間が一次元の場合、図20に示すように、長さ L の等間隔の5つの領域に分割し、各領域に位置する粒子をそれぞれクラス $C(1)$ 、 $C(2)$ 、...、 $C(5)$ に属するとする。 $C(j)$ の境界位置を $b(j)$ 、 $b(j+1)$ とし、粒子の位置座標が $Y(i)$ で表されるとすると、 $b(j) \leq Y(i) < b(j+1)$ ならば粒子が属するクラスは $i \in C(j)$ ということになり、一意に決定されるので、各クラスの所属粒子に関する情報は容易に決定できる。また、位置座標の更新により所属粒子を変更する事も容易である。この情報を記憶しておけば、あるクラスにどのような粒子が所属しているかがすぐにわかる。そこで成分 i の処理事項として所属クラス $C(j)$ を記憶しておく。もし、相互作用が及ぶ距離が L であるとする、自分が属するクラスと隣り合うクラス以外のクラスに属する粒子との相互作用を計算する必要はなく、隣り合うクラスに属する粒子のみを考慮して計算すればよい。従って、最重要成分として成分 i が演算処理される場合には、3つのクラス $C(j-1)$ 、 $C(j)$ 、 $C(j+1)$ に属する成分のみを対象として計算すればよい。このことは三次元空間の場合にも拡張できる。その場合、粒子の三次元座標を表す三次元ベクトルを一つの成分とみなして処理する。

【産業上の利用可能性】

【0041】

本発明を利用する計算処理問題としては以下の事例が考えられる。多数のパラメータを含むエネルギー関数を最小化するような最適化問題において準最適解を求める問題の数値解法。熱揺らぎのある溶媒中におけるタンパク質の一次構造から立体構造が形成されるフォールディングのシミュレーション。確率の変動のある大規模ネットワーク上の部分的信号伝搬のシミュレーション。以上のように、確率的な変動があっても、状態値が急激に発散することなく、最終的にある収束状態の付近に落ち着く傾向のあるシステムのシミュレーションを行なう場合に、効率良く、低コストで結果を得ることが出来る。

【図面の簡単な説明】

【0042】

【図1】 本発明の第1の実施形態を示す基本ブロック構成図

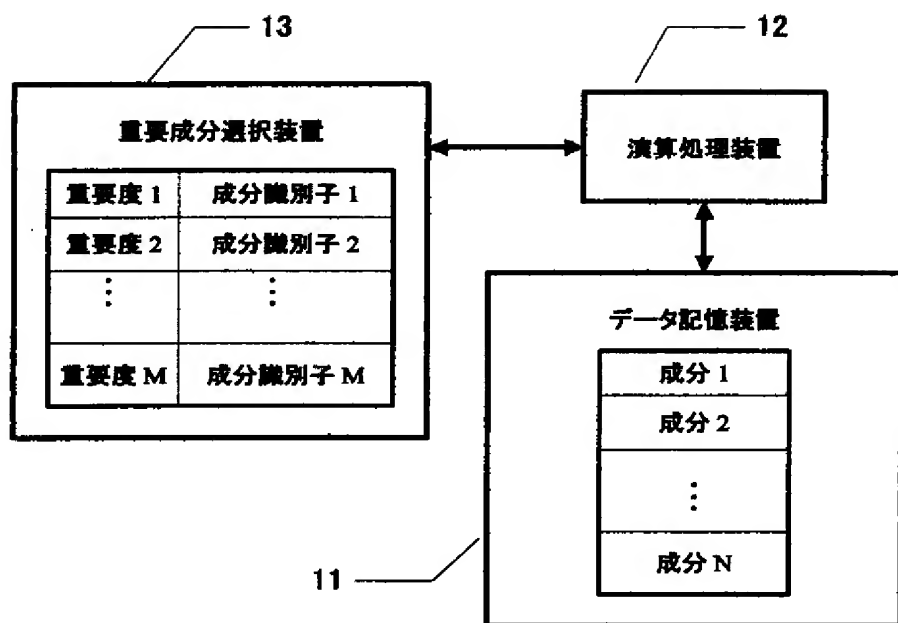
【図2】 第1の実施形態の動作の概要説明図

- 【図 3】 本発明の第 2 の実施形態を示す基本ブロック構成図
【図 4】 第 2 の実施形態の動作の概要説明図
【図 5】 本発明の第 3 の実施形態を示す基本ブロック構成図
【図 6】 第 3 の実施形態のためのデータ記憶装置の一例を示すブロック回路図
【図 7】 本発明の第 4 の実施形態である重要成分選択装置用ハードウェアの基本ブロック構成図
【図 8】 第 4 の実施形態の基本回路構成の具体例を示すブロック回路図
【図 9】 第 4 の実施形態の比較器の一例を示すブロック回路図
【図 10】 第 4 の実施形態のバイパス配線用比較器の一例を示すブロック回路図
【図 11】 第 4 の実施形態のバイパス配線の具体例による説明図
【図 12】 第 4 の実施形態のバイパス配線方法の一例に関する説明図
【図 13】 第 4 の実施形態のセレクタ・演算処理器の一例を示すブロック回路図
【図 14】 第 4 の実施形態のセレクタ・演算処理器の制御信号と出力の関係表
【図 15】 第 4 の実施形態の P U S H の動作例を示す説明図
【図 16】 第 4 の実施形態の P O P の動作例 1 を示す説明図
【図 17】 第 4 の実施形態の P O P の動作例 2 を示す説明図
【図 18】 成分データの記憶と演算処理方法に関する本発明の第 5 の実施形態図
【図 19】 成分データの記憶と演算処理方法に関する本発明の第 6 の実施形態図
【図 20】 第 6 の実施形態における成分データと処理事項の関係の説明図

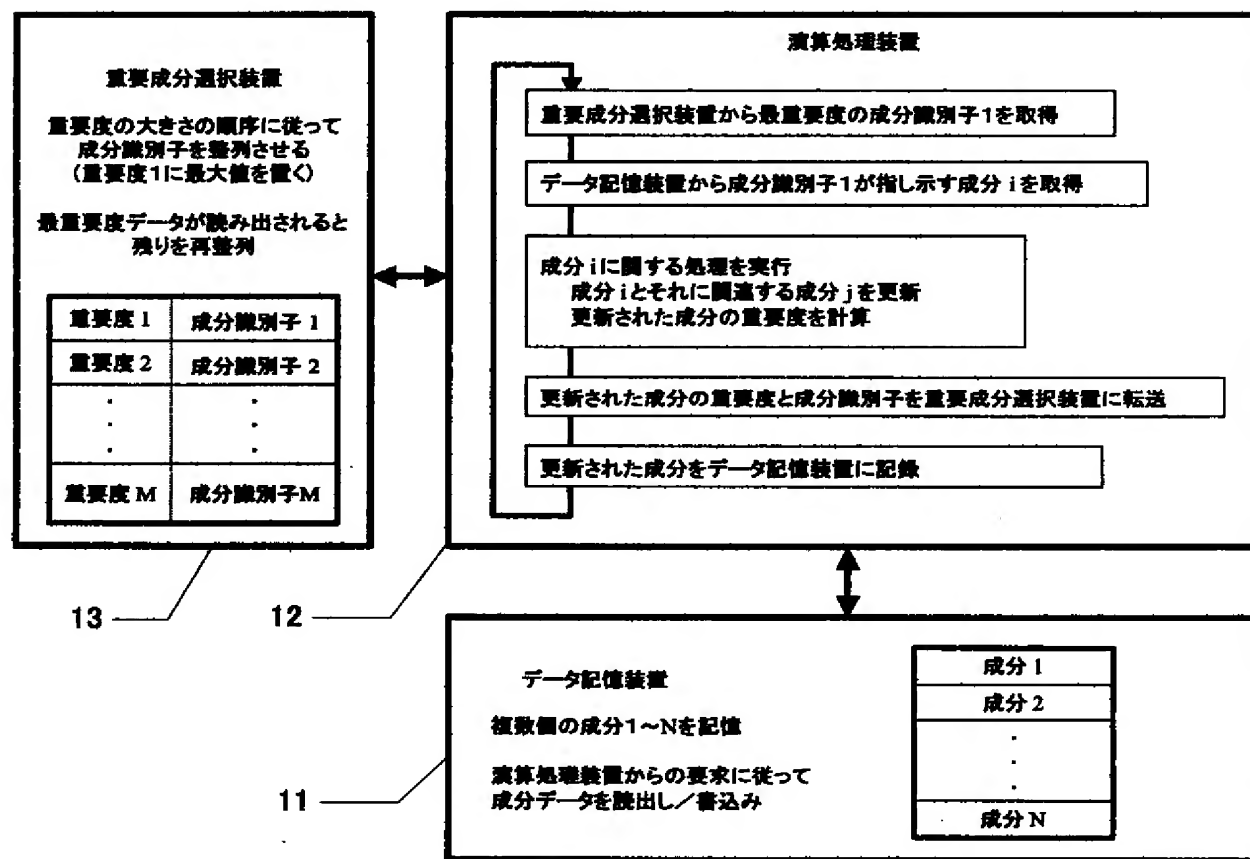
【符号の説明】

【 0 0 4 3 】

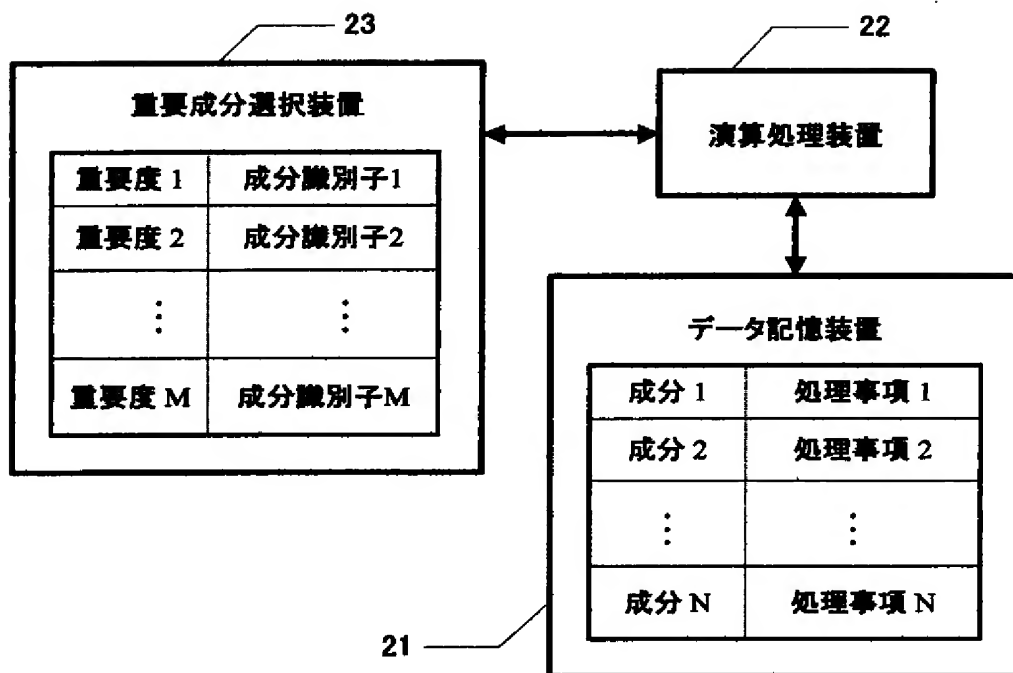
- 1 1 , 2 1 , 3 1 , 4 1 , 5 1 , 6 1 データ記憶装置
1 2 , 2 2 , 3 2 , 4 2 , 5 2 , 6 2 演算処理装置
1 3 , 2 3 , 3 3 , 4 3 , 5 3 , 6 3 重要成分選択装置
3 4 メモリ入出力制御回路
3 5 データ転送制御回路
3 6 F I F O バッファレジスタ
3 7 後書き込み制御回路
3 8 記憶素子
3 9 データセレクタ
4 1 0 レジスタ
4 2 0 データ整列制御回路
4 2 1 比較器
4 2 2 セレクタ・演算処理器
4 2 3 大小比較器
4 2 4 一致比較器
4 2 5 データセレクタ
4 2 6 演算処理器
4 2 7 出力セレクタ



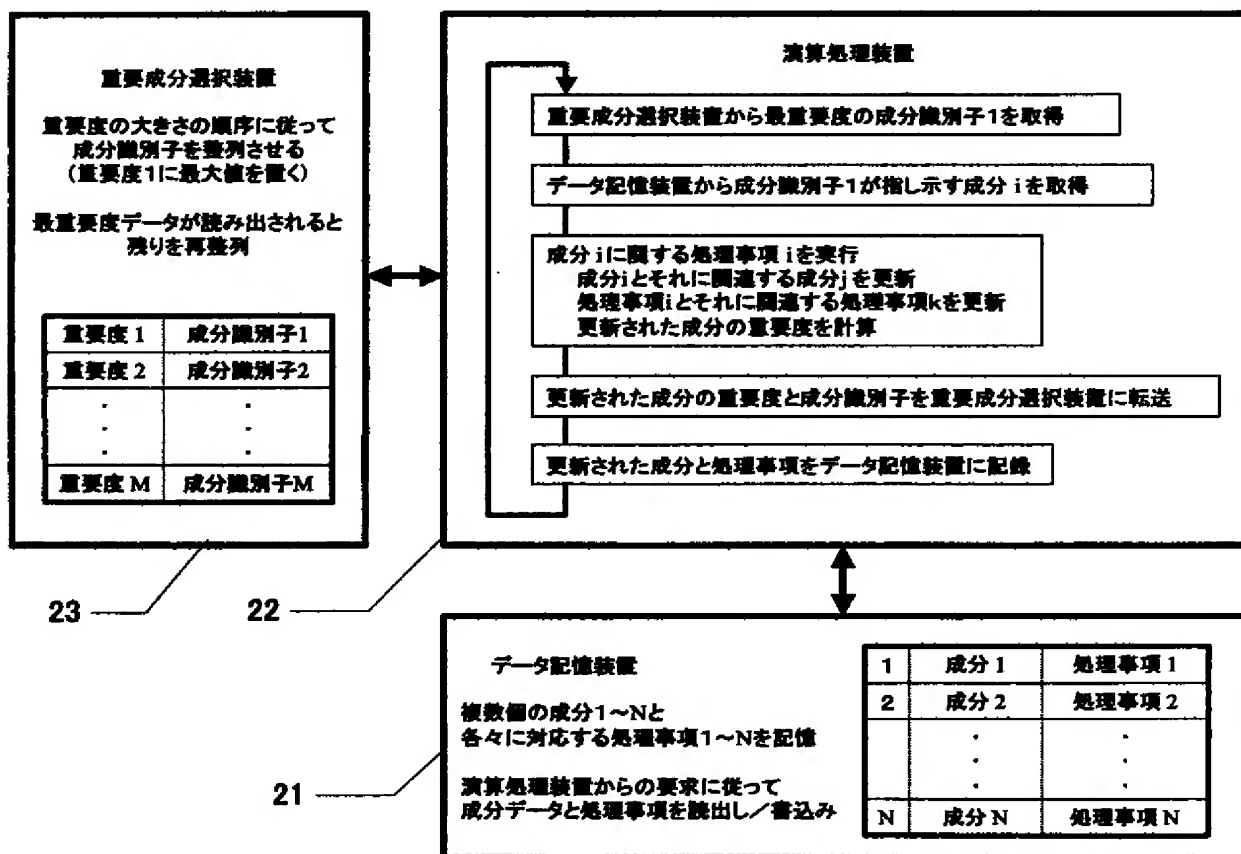
【図 2】



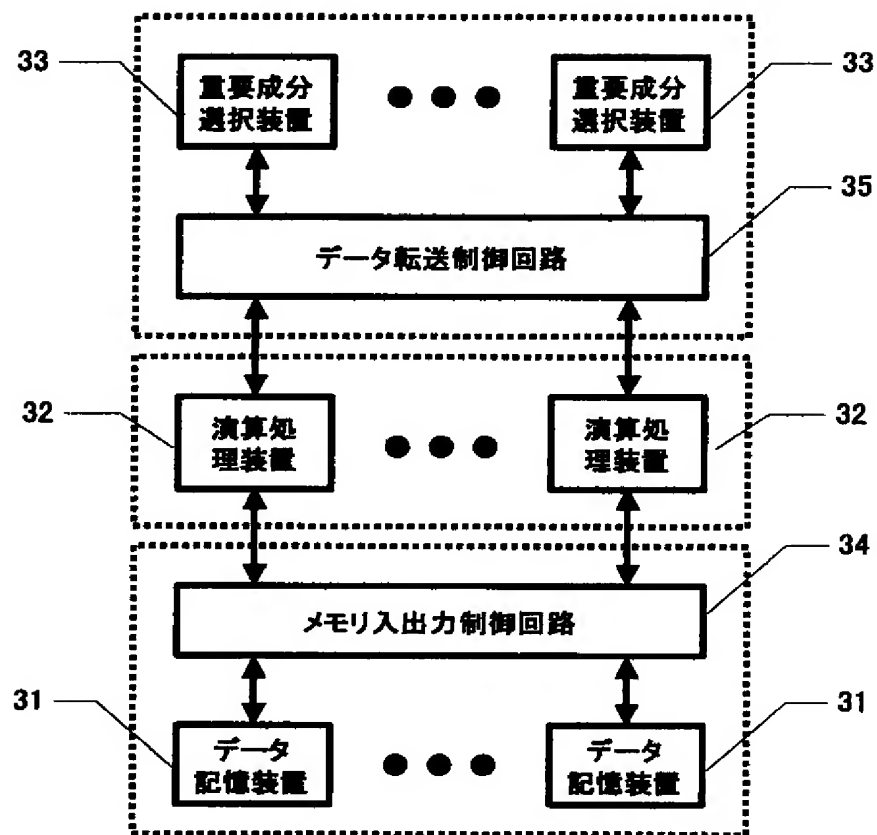
【図 3】



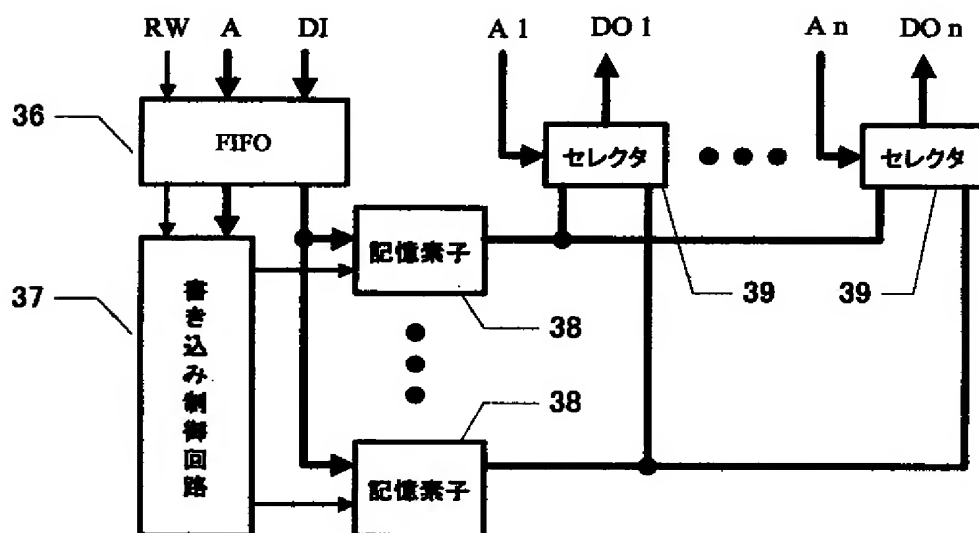
【図 4】



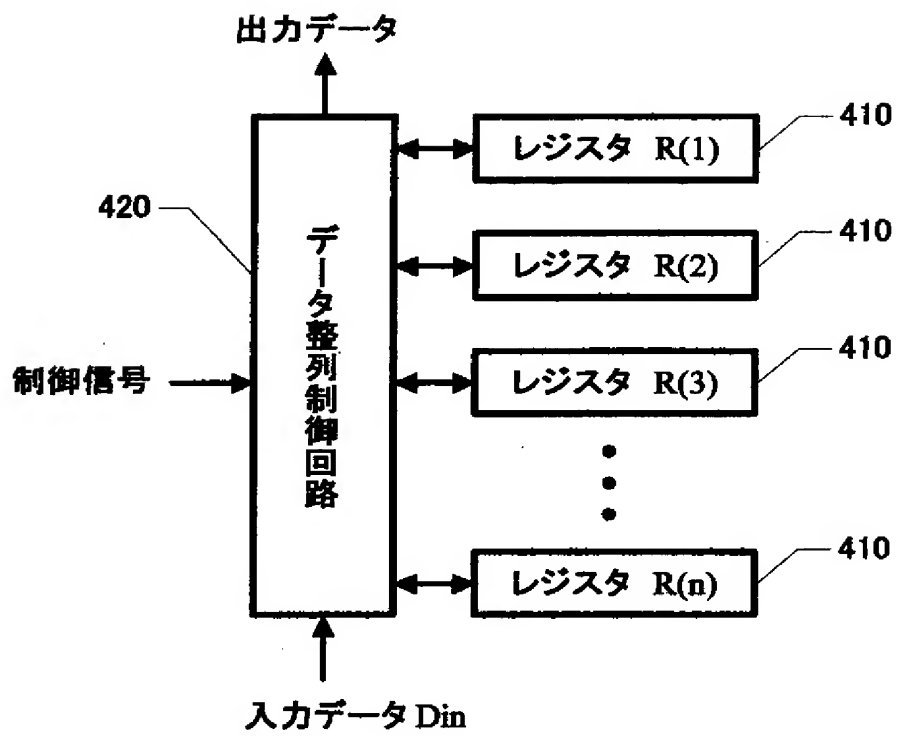
【図 5】



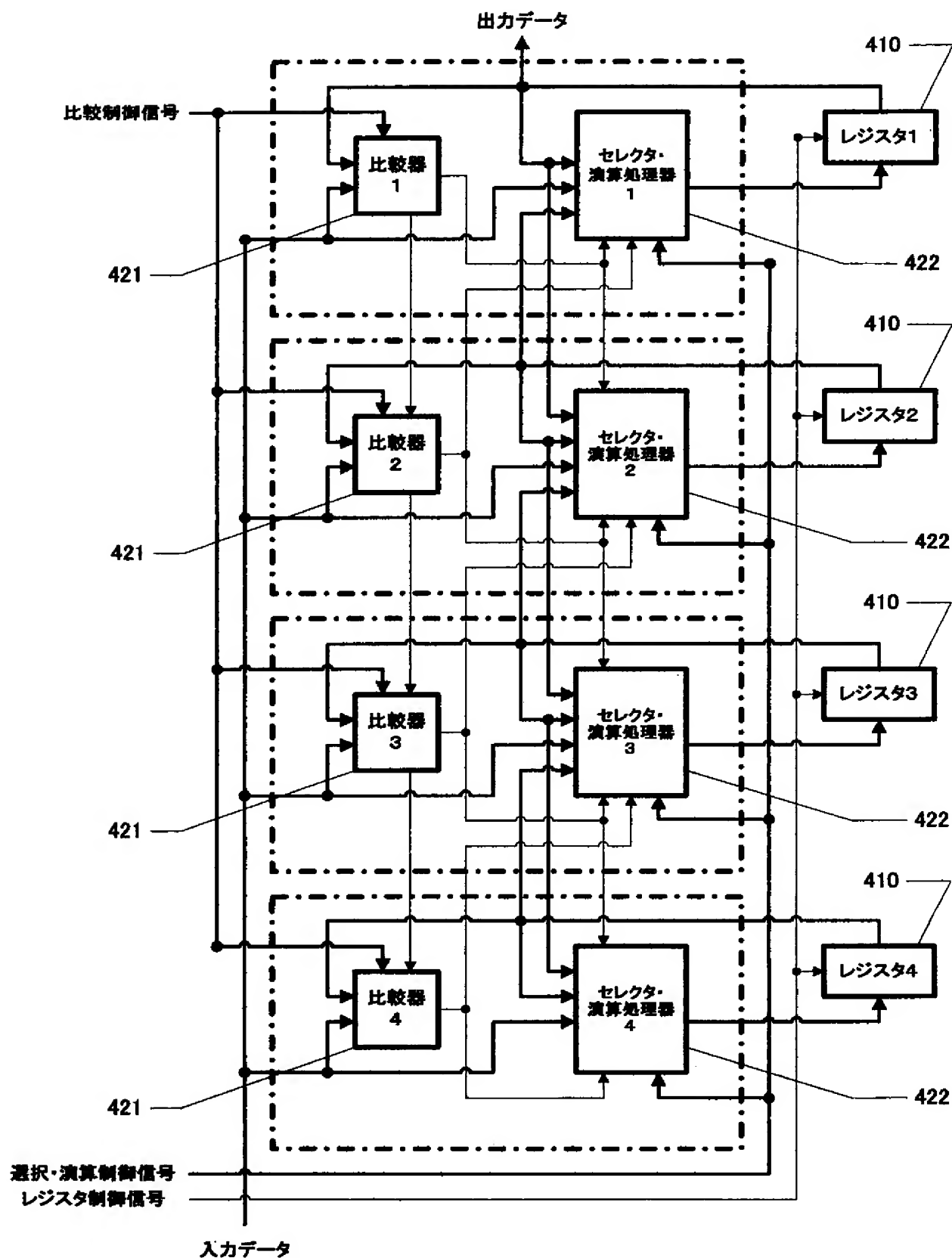
【図 6】



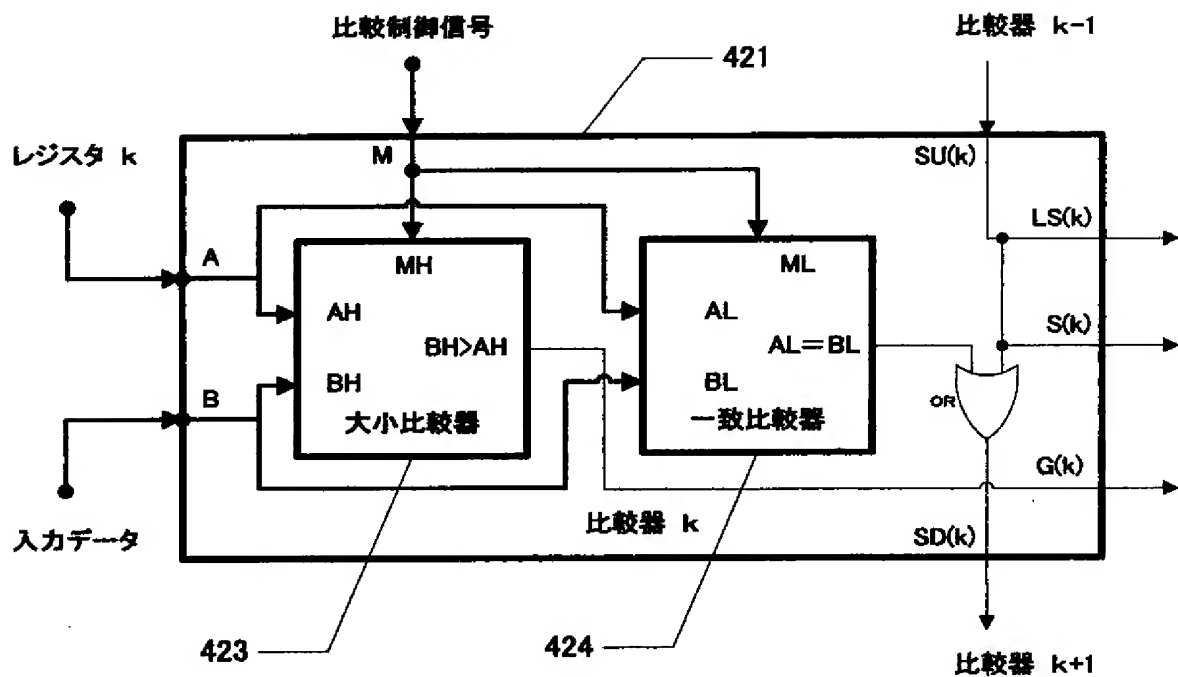
【図 7】



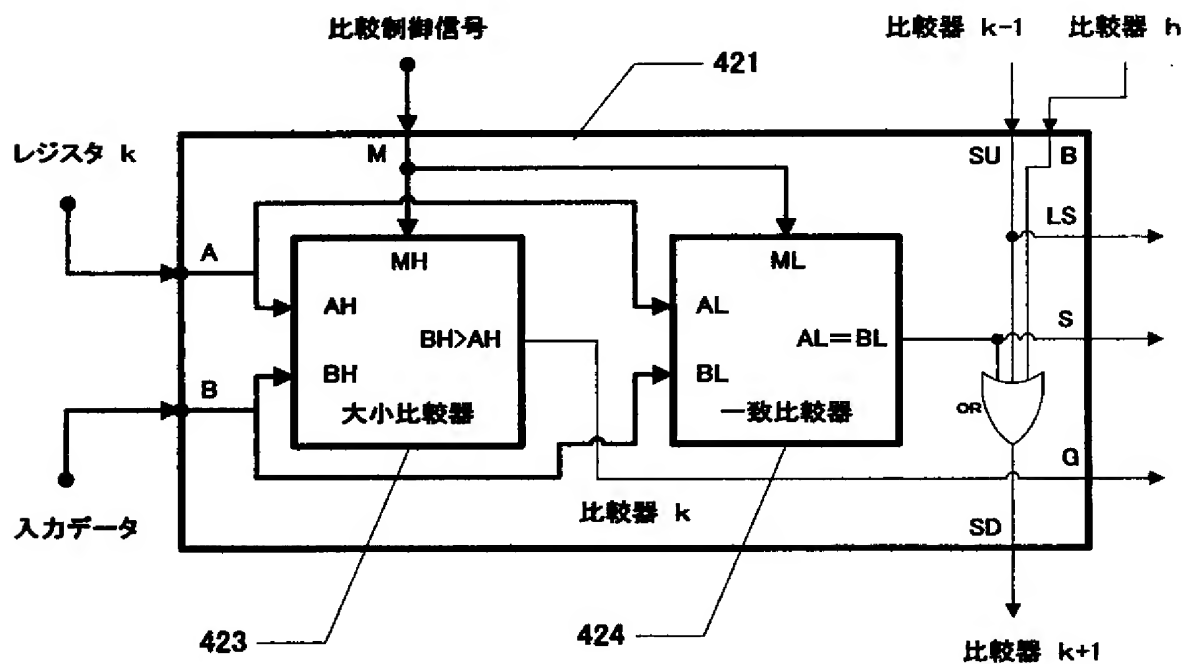
【図 8】

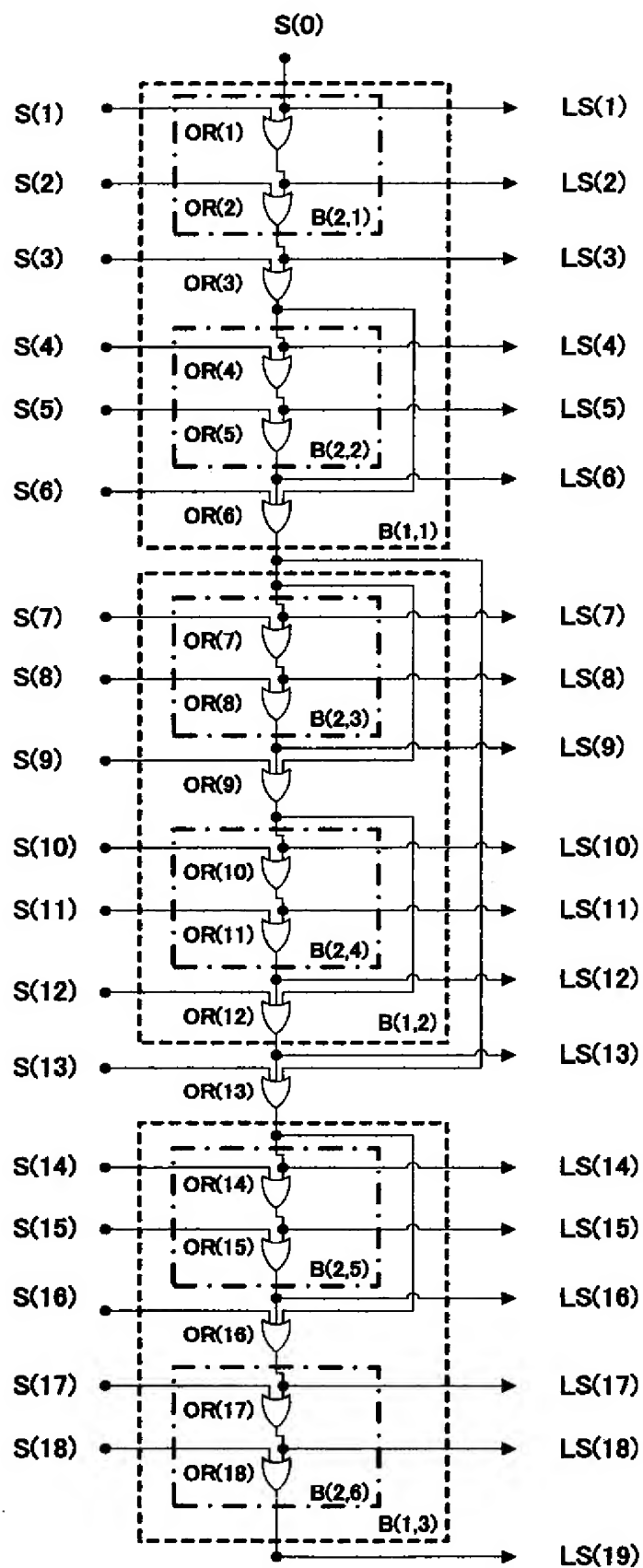


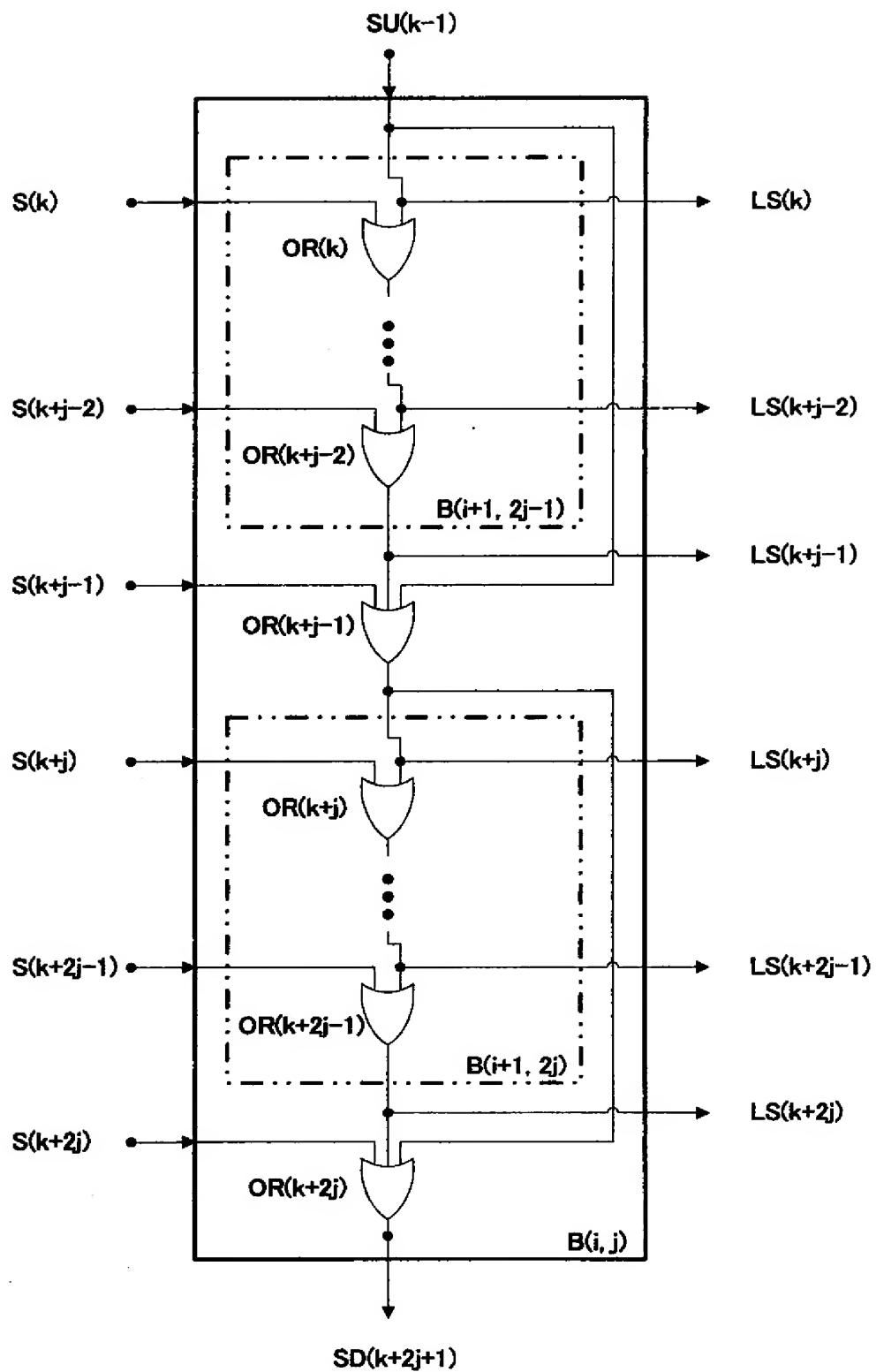
【図 9】



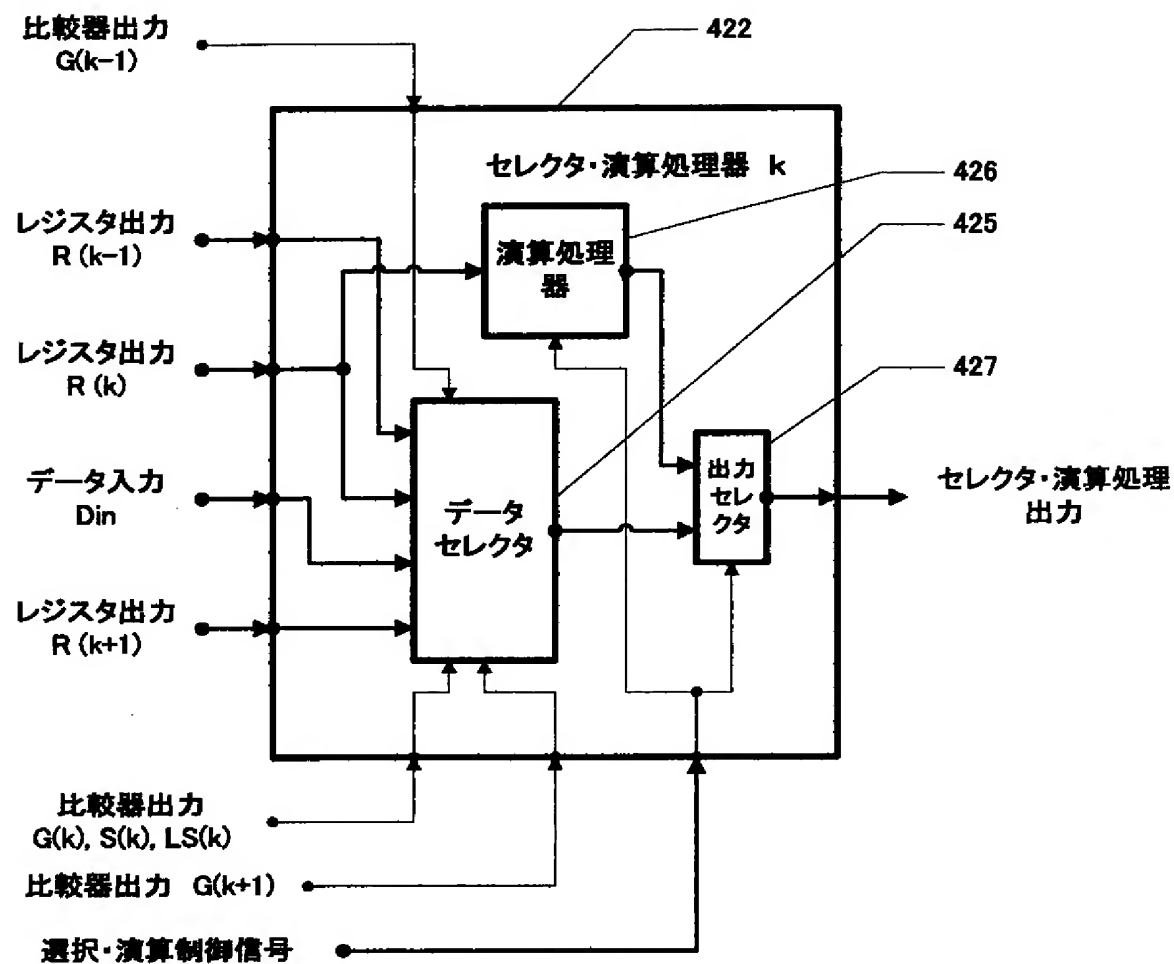
【図 10】







【図 1 3】



【図 1 4】

制御信号	条件		比較器出力					セクタ・演算処理出力
	重要度	同一成分順位	G(k-1)	G(k)	G(k+1)	S(k)	LS(k)	
PUSH	$P(k) \geq P_{in}$	*	*	0	*	*	*	R(k)
	$P(k-1) \geq P_{in} > P(k)$	$k \leq K_s$	0	1	*	*	0	Din
	$P_{in} > P(k-1) \geq P(k)$	$k \leq K_s$	1	1	*	*	0	R(k-1)
	*	$k > K_s$	*	*	*	*	1	R(k)
POP	$P(k) \geq P(k+1) \geq P_{in}$	*	*	0	0	*	*	R(k+1)
	$P(k) \geq P_{in} > P(k+1)$	$k < K_s$	*	0	1	0	0	Din
	$P_{in} > P(k)$	$k < K_s$	*	1	*	0	0	R(k)
	*	$k = K_s$	*	*	*	1	*	R(k+1)
	*	$k > K_s$	*	*	*	*	1	R(k+1)
ADD	*	*	*	*	*	*	*	$P(k) + a$
SFTL	*	*	*	*	*	*	*	$P(k) \ll a$
SFTR	*	*	*	*	*	*	*	$P(k) \gg a$

【図 1 5】

PUSH

k	G	大小比較	重要度データ	成分識別子	一致比較	S	LS
1	0	$P_{in} \leq$	98	777	\neq	0	0
2	0	$P_{in} \leq$	74	200	\neq	0	0
3	0	$P_{in} \leq$	60	101	\neq	0	0
4	1	$P_{in} >$	33	020	\neq	0	0
5	1	$P_{in} >$	27	700	\neq	0	0
6	1	$P_{in} >$	15	555	$=$	1	0
7	1	$P_{in} >$	8	300	\neq	0	1
8	1	$P_{in} >$	5	808	\neq	0	1

$K_s \rightarrow$ (row 6)

60	555
----	-----

入力データ

【図 1 6】

POP

k	G	大小比較	重要度データ	成分識別子	一致比較	S	LS
1	0	Pin ≤	98	777	≠	0	0
2	0	Pin ≤	74	200	≠	0	0
3	0	Pin ≤	60	101	≠	0	0
4	1	Pin >	33	020	≠	0	0
5	1	Pin >	27	700	≠	0	0
6	1	Pin >	15	555	=	1	0
7	1	Pin >	8	300	≠	0	1
8	1	Pin >	5	808	≠	0	1
			0	0			
			60	555			

入力データ

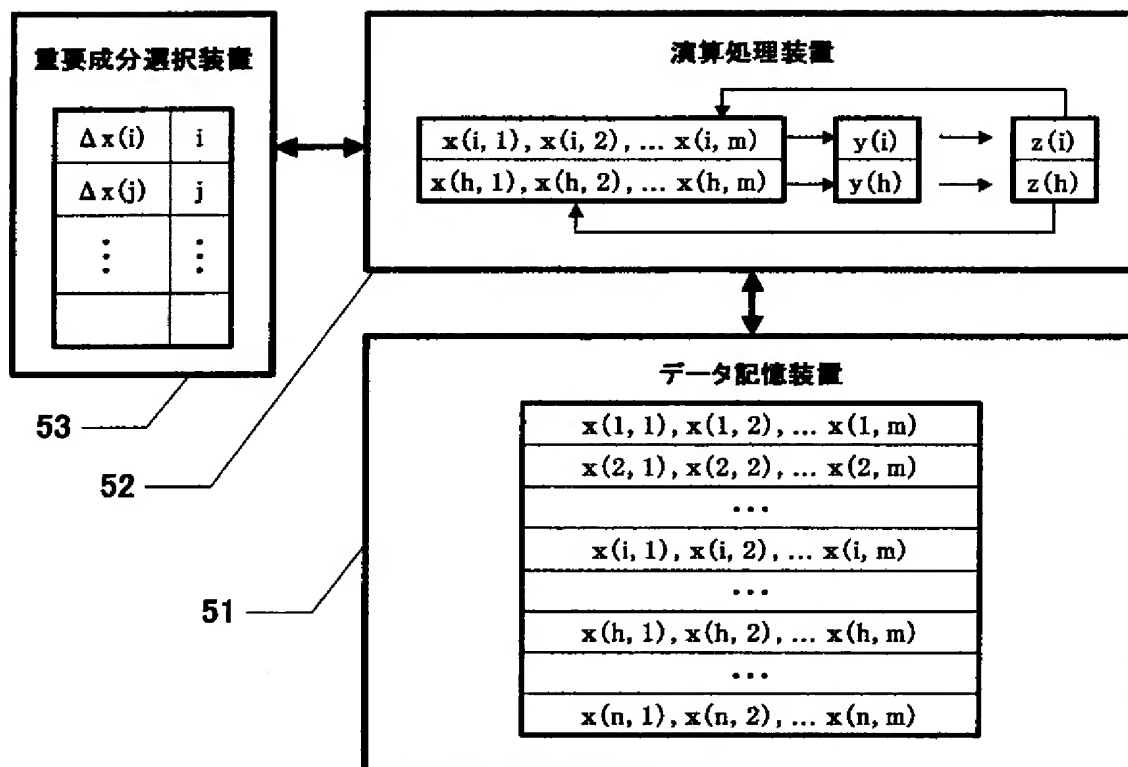
【図 1 7】

POP

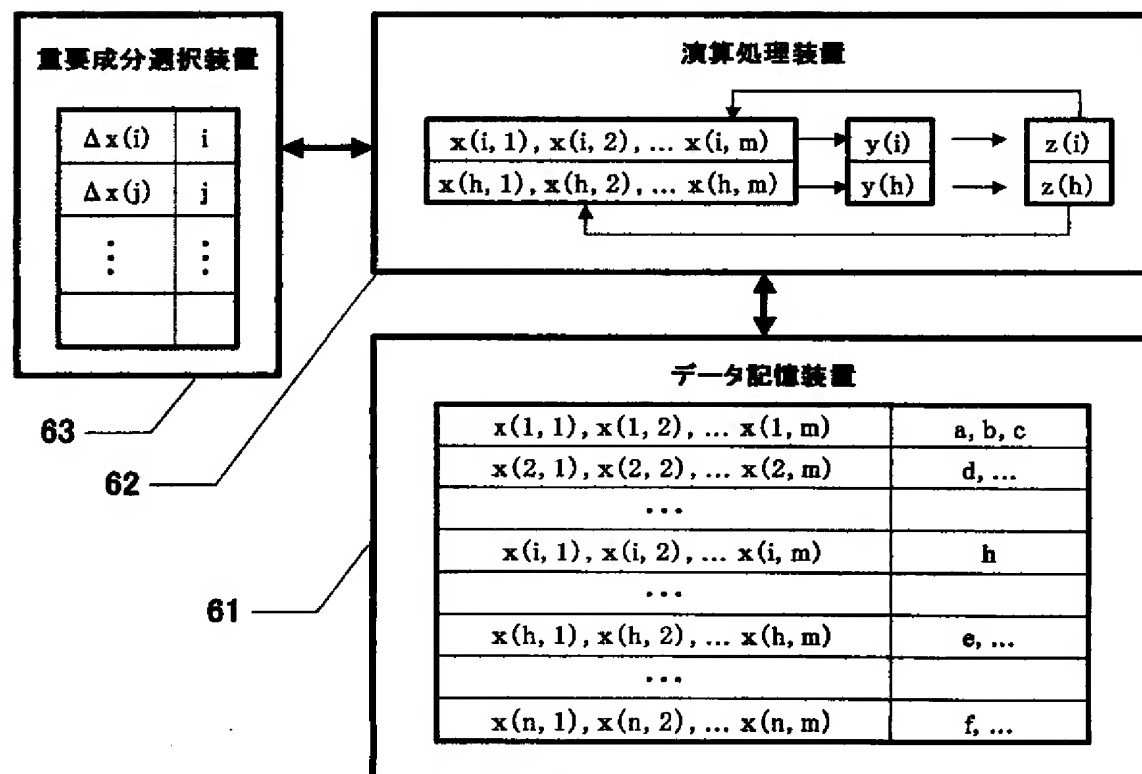
k	G	大小比較	重要度データ	成分識別子	一致比較	S	LS
1	0	Pin ≤	98	777	≠	0	0
2	0	Pin ≤	74	200	≠	0	0
3	0	Pin ≤	60	101	≠	0	0
4	0	Pin ≤	33	020	≠	0	0
5	0	Pin ≤	27	700	≠	0	0
6	0	Pin ≤	15	555	=	1	0
7	0	Pin ≤	8	300	≠	0	1
8	1	Pin >	5	808	≠	0	1
			0	0			
			7	555			

入力データ

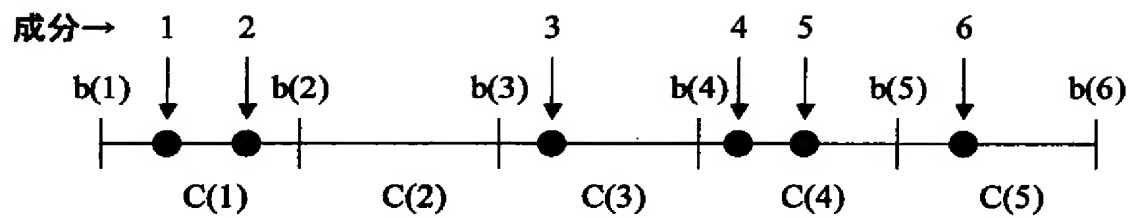
【図 18】



【図 19】



【图 2 0】



$C1=\{1, 2\}, C2=\{\}, C3 = \{3\}, C4 = \{4,5\}, C5 = \{6\}$

$i \in C(j) \text{ for } b(j) \leq Y_i < b(j+1)$

【書類名】 要約書

【要約】

【課題】 従来、物理現象や社会現象など多数の成分からなるシステムのシミュレーションを行なう場合、全ての成分について状態値を少しずつ更新することにより正しい数値を求める計算を行っていた。そのため、成分数が増加すると膨大な計算時間を要し、結果を得る事が困難になるという欠点があった。特に、全成分の内のほんの一部分のみが変化する場合に、ほとんど変化しない成分についても無駄に計算時間を消費したり、大きなノイズを含むシステムではもともとの数値データの精度が低いにもかかわらず必要以上の高精度な計算を行なったりするなど、計算資源の無駄遣いが多かった。

【解決手段】 シミュレーション計算において重要度の高い成分を選び出し、その重要度の高い成分から優先的に計算処理を行い、計算対象となった成分の重要度を再評価して、その結果をその後の重要成分の選択に反映させる。重要度の低い成分に対する計算は後回しにするか、または一定の基準で無視することにより、できるだけ早く効率よく概算結果を獲得し、その後は必要に応じて時間をかけて計算精度を向上させる。

そのため、各成分の状態値の計算以外に各成分の重要度を計算する過程を加え、それ进行处理するための専用装置として重要成分選択装置を設け、その装置によって各成分をその重要度に応じて順位付ける。また、各成分の状態値の計算においては重要成分選択装置の指示する最重要成分を優先して計算処理を行なう。

重要成分選択装置において最高速の処理を要求される計算モデルに対しては、重要成分の順位付け計算をハードウェアで実現する。また、変化の少ない成分に関する無駄な演算処理を省略するため、記憶装置において複数個の成分データのの一つ一つに対して複数個の候補値データを保持するものとする。演算処理装置ではその複数個の候補値の統計量を成分データの代表値とみなし、各成分データの代表値について計算処理を行ってその更新値を算出し、その更新値と最も大きく異なる候補値をその更新値で置き換える。この場合の統計量としては平均値や中央値などを用いる。そして、重要成分選択装置では候補値のばらつきの大きさをその成分データの重要度とみなして、候補値のばらつきの大きさの順位で重要成分を選択する。候補値のばらつきの大きさはシミュレーションにおいて成分データの値が十分に収束していない状況や時間変化が大きい状況を表すので、変動の少ない成分データよりも変動の大きい成分データの計算処理を優先させることができる。これにより、変化の少ない計算過程を省略することができ、より高速に計算結果を得ることが可能になる。そのまま必要に応じて計算時間を延長する事で更に精度を上げる事も出来る。

【選択図】 図 1

出願人履歴

3 0 4 0 1 9 2 3 0

20040330

新規登録

奈良県奈良市朱雀3丁目15番地1号 5－407号室

藤田 修